On Two New Trends in Evolvable Hardware: Employment of HDL-based Structuring, and Design of Multi-functional Circuits

Adrian Stoica
Ricardo S. Zebulum
Didier Keymeulen
M. I. Ferguson
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109
adrian.stoica@jpl.nasa.gov

Xin Guo
Chromatech, Alameda CA 94501

Abstract

This paper comments on some directions of growth for evolvable hardware, proposes research directions that address the scalability problem and gives examples of results in novel areas approached by EHW. The directions of growth include Software/Hardware hybrids, electronic/non-electronic hybrids, and networked systems. The research directions proposed here are 1) evolutionary compilation of descriptions from behavioral Hardware Description languages (HDL) to structural HDL (for both the case of digital and analog/mixed signal) 2) evolutionary synthesis, i.e. converting from synthesizable HDL to circuits and 3) hardware-software partitioning (co-design) for CPU/FPGA hybrids. The results presented here illustrate evolutionary design of multi-functional/adaptive circuits including polymorphic and reconfiguration based circuits, and evolution of optimized circuits, in particular low-voltage circuits.

1. EHW directions of growth and immediate challenges

Evolvability may become a key characteristic of infrastructures of 2020 and beyond. After the fixed and then the reconfigurable hardware generation, the next one will be self-configurable and evolvable. Evolvable hardware technology will grow at least in three directions: 1) toward including software and evolving as hybrid-ware (the distinction between hardware and software is expected to blur, at least as far as the schism in system development is concerned), 2) toward including non-electronics, such as antennas, MEMS or biological systems, and 3) toward using parallel evaluations of populations in a network environment. "Smart materials" and distributed, high bandwidth, sensing structures will embed small areas of silicon or another material, containing up to a few hundred programmable transistors or other active devices in areas smaller than 10x10 micron, acting as tiny signal processors co-located with miniature sensing/actuation devices. These would provide local adaptive information processing, for example, in "smart skins" for aircraft or submarines.

Two most important (interrelated) challenges for evolvable hardware are scalability and the need for complete upfront formal specifications, e.g. in terms of a language. So far only relatively simple systems have been evolved. The components used are primitive elements, for example device-level (transistor, capacitor, resistor) for analog or gate level for digital. For any complex system the number of components used may be relatively large and the total number of ways of interconnecting them, and consequently the size of the search space for a solution is huge. The space can be reduced by keeping an acceptable scope of focus by using higher-level building block. However it is unclear how exactly to select and little progress has been achieved through encapsulating such sub-circuits obtained during evolution (corresponding to Automatically Defined Functions). The need for upfront complete specifications is reflected in situations when in order to evolve a gate, timing specifications were needed for more than one time domain (see examples in [1]), yet,

Proceedings of the 2002 NASA/DOD Conference on Evolvable Hardware (EH'02)
0-7695-1718-8/02 $17.00 © 2002 IEEE

testing devices in a large number of situations for which we want to guarantee functionality is inefficient.

2. Evolving from high-level specifications

- *From behavioral HDL to circuit through synthesizable HDL.*

One way to approach the scalability problem is to first admit that what we address is an open problem for both analog *and* digital (and for system design in general). The perception in the evolvable hardware community exists that the digital automated design/synthesis problem is solved by current techniques and tools. What is missed is that only *structural* VHDL (Verilog) is synthesizable, while *behavioral* VHDL is not. The reason is simple: structural VHDL offers a problem decomposition! Thus the tools only have to deal with implementation of a simpler block, and also the set of library elements offers easy/direct matches. (The boundary between behavioral and structural depends on the vendor supported language/extension and size of IP library, etc). However, evolutionary design is behavior-oriented and poorly-specified in form of response curves instead of using standard language such as HDL. In this context the following two research directions appear promising:

a) *Evolutionary-based compilation of behavioral HDL to structural HDL* (analog or/and digital). Force specifications to be in a standard behavioral language.

b) *Evolutionary-based synthesis of structural AHDL*. Structural AHDL may be the required first step to automatic analog synthesis. The building blocks may be sufficiently small to allow evolution to find optimal solution.

We believe that by decomposing the problem first into a *functional to structural* translation and then a *structural to primitives* one, the chances of evolution to approach complex system are much improved.

-*Hardware/software evolutionary co-design for FPGA/CPU hybrids and other Systems-on- a-Chip*

This direction is especially timely in the context of the embedded systems industry rapidly moving toward integration of programmable and reconfigurable devices, with a powerful convergence toward hybrid FPGA/CPU architectures. Ultimately these will use flavors of on-chip hybrids such as the new Xilinx Virtex II Pro chip. There are no tools allowing designers to go from system level specification, e.g., in Matlab/Simulink and convert it to an efficient hardware/software allocation/partitioning.

3. Evolving circuits satisfying multiple and special constraints

A variety of novel circuits that satisfy multiple constraints may be obtained through evolution. The multiple constraints discussed here are multiple functions superimposed on the same fixed (without switches) circuit and multiple functions on reconfigurable architectures (possibly with minimal switches to reconfigure at functional changes). Special constraints exemplified here relate to power/voltage minimization.

Polymorphic circuits (introduced in [2]) are circuits with multiple superimposed functionality, where function changes not as a result of switches (non-existent) but due to changes in the operational point of components. These are new type of circuits and there are no design guidelines for humans on how to design such circuits, since they are not based on the traditional block decomposition and functional modularity. An example of an evolved circuit is shown in Figure 1. The circuit shown in this figure behave as an AND gate for V_{DD}=1.2V and as an OR gate for V_{DD}=3.3V.

An example of a reconfigurable multi-functional circuit mapped on programmable array is shown in Figure 2, where it is shown the response of narrow band-pass filters with center frequencies at different locations (5 and 25kHz). The reconfigurable circuit can map a variety of filters. Evolution is used to synthesize several filters based on the same set of available resources. An extra optimization step that can be added is to minimize the number of architectural changes, i.e. the number of switches needed to be changed from topology to another. This would reduce the time to reconfigure since usually the bits are changed serially (at least for larger devices) or in clusters (which can be the subject of minimization procedure as well).

An example of low voltage design is illustrated in Figure 3. The NAND circuit evolved for a value of V_{DD} of 0.8V is outperforming a conventional NAND circuit in terms of speed. In this example the connection of the substrate was allocated by the evolutionary algorithm, leading to an unusual topology.

4. Conclusions

New approaches are needed in order to reach the full potential of evolvable systems. Scalability and completeness of specifications are primordial. It is proposed to approach these through hardware description languages, formulating the requirements/ specifications in HDL. It is also proposed to use intermediate, structural level representation, and thus employ evolution in two phases: behavioral to structural, and structural to circuits. Evolution proved efficient in the design of circuits satisfying multiple constraints. We illustrated multi-functional designs in polymorphic circuits (promising since there are no human design guidelines for such circuits) and on reconfigurable architectures (a new field that will expand –for analog- with availability of programmable analog arrays and in digital if fast reconfiguration is sought by minimal distance between

two solutions) and special constraints in a low-power circuit (letting evolution explore new unconventional design solutions, such as the allocation of substrate, etc).

Acknowledgements

The research described in this paper was performed at the Center for Integrated Space Microsystems, Jet Propulsion Laboratory, California Institute of Technology and was sponsored by the Defense Advanced Research Projects Agency (DARPA) and the National Aeronautics and Space Administration (NASA).

References

[1] Stoica et Al., "Evolving Circuits in Seconds: Experiments with a Stand-Alone Board Level Evolvable System" 2002 NASA/DoD Conference on Evolvable Hardware, Alenxandria, VA., July, 2002 (to be published).
[2] Stoica, A. Zebulum R. and Keymeulen D. "Polymorphic Electronics" International Conference on Evolvable Systems, October 2001, Springer-Verlag, pp.291-302, Tokyo, Japan.

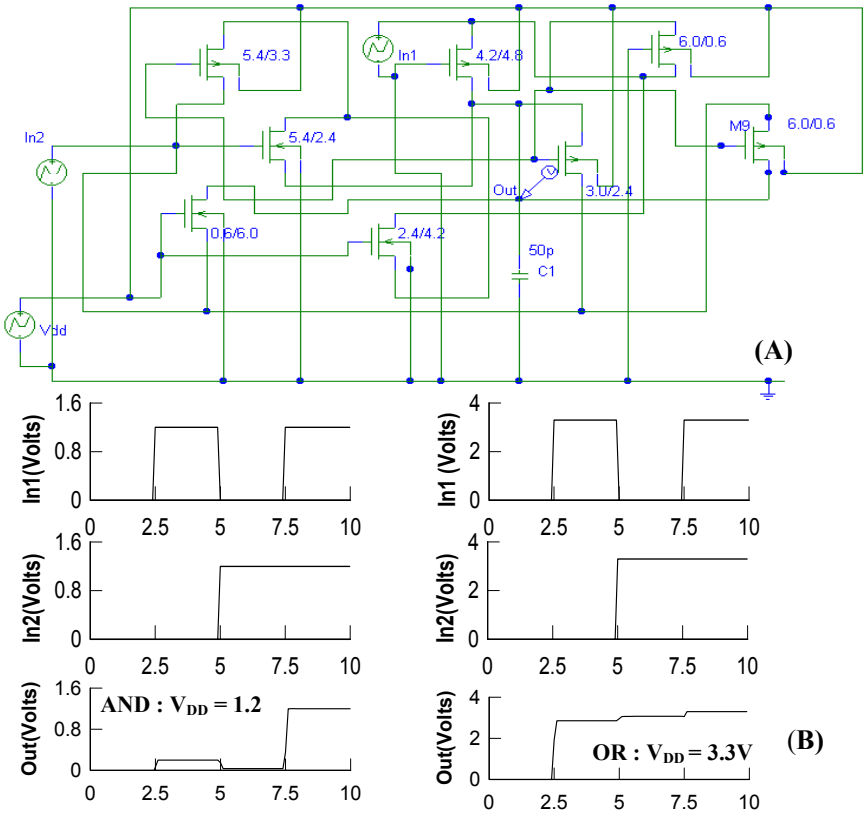

Figure 1. VDD controlled Polymorphic circuit and response. Schematic of the polymorphic circuit controlled by supply voltages (A). Circuit inputs and response for two cases, V_{DD}=1.2V (left) and V_{DD}=3.3V (right). Axis X of the graphs gives the time in milliseconds.

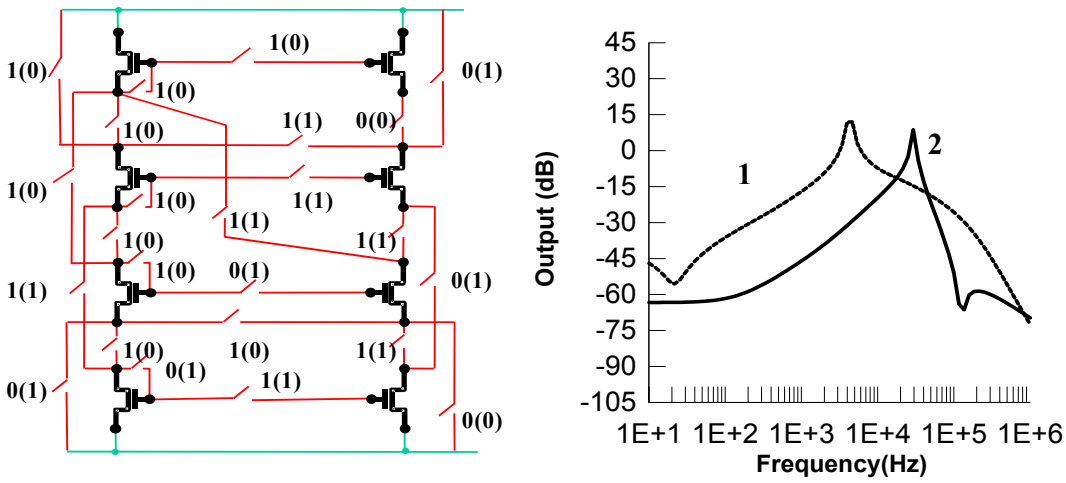

Figure 2. Filter on re-configurable circuit. Simplified architecture at the left and response at the right. (capacitors omitted in the figure). The binary state of the switches is represented next to the respective switch for the two filters, filter 2 between brackets. Filter 1 presents a gain of 11dB at 5kHz and roll-off about –30dB/dec. Filter 2 presents a gain of 9dB at 25kHz and roll-off about –40dB/dec for the lower band and –70dB/dec for the upper band.

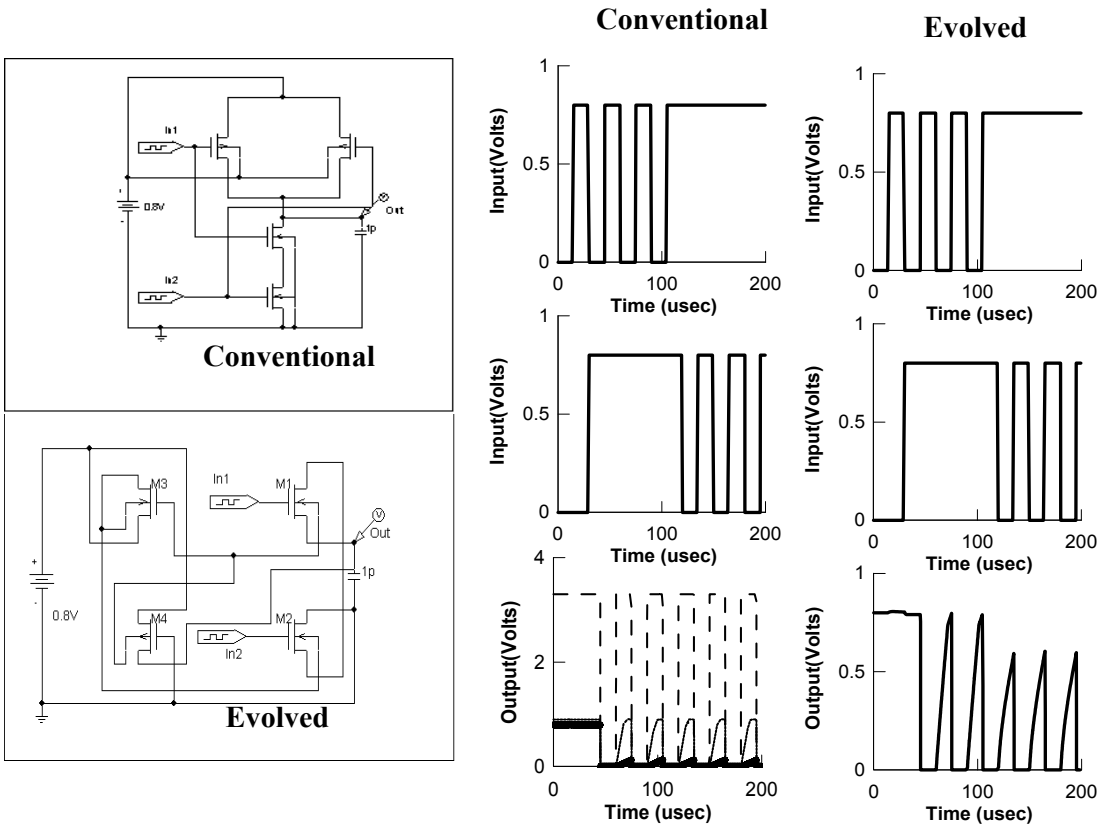

Figure 3. Low-voltage circuit and response – comparison with conventional circuit (left) and response (right). Output of conventional circuit shown for three power supply values: 0.8V (full line), 0.9V (dots/thin line) and 3.3V (traces). Note degradation in performance of conventional circuit for 0.8V. Output of evolved circuit (shown for 0.8V) stays at the correct logic levels when inputs are toggled.